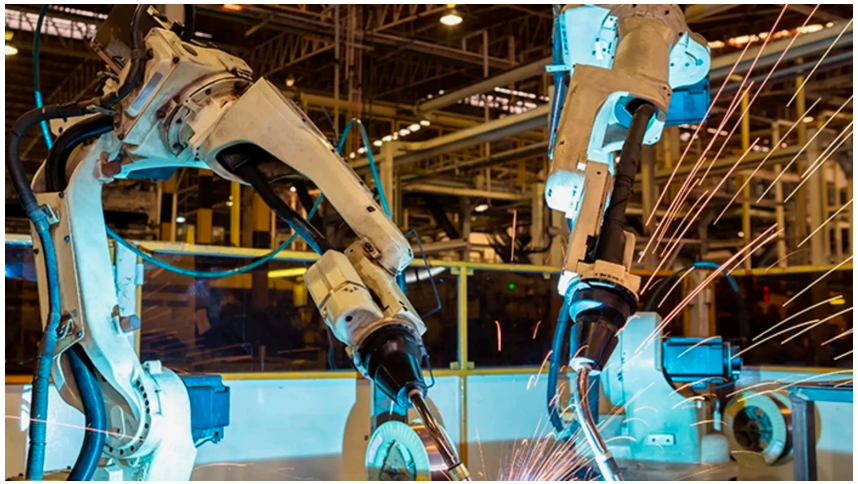
Neural Stochastic Control and Verification for Safe Autonomy

Djordje Zikelic

Nanyang Technological University, Singapore

Symposium on AI Verification 2026

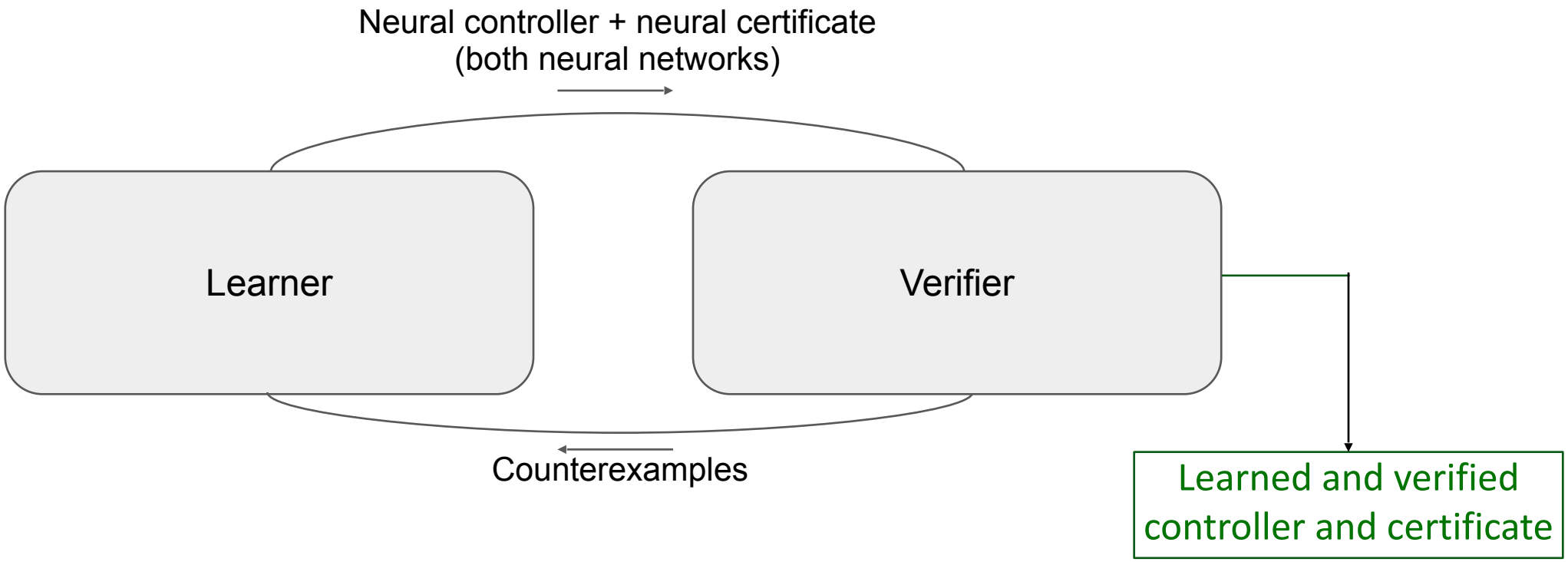
Learning-enabled and neural control



- Reinforcement learning
- Other learning-based methods (e.g. supervised and unsupervised learning)
- Program synthesis (e.g. programmatic RL)

Safe autonomy requires **correctness guarantees**

Neural control with certificates



Idea: Learn and verify neural certificate for the specification
(A certificate is a locally checkable witness of correctness)

Some examples of certificates

- **Lyapunov functions** (continuous-time stability)
- **Ranking functions** (discrete-time reachability)
- **Barrier functions** (discrete/continuous-time safety)

Verification via reduction to SMT solving [Chang et al. 2019, Abate et al. 2021, Zhao et al. 2020]
or interval/linear bound propagation [Mathiesen et al. 2023, Hu et al. 2024, Vertovec et al. 2025]

Several challenges still remain

Challenge 1

How to learn neural controllers for temporal specification tasks?

(RL struggles with long-horizon and complex specification tasks)

Challenge 2

How to verify neural controllers in the presence of environment uncertainty?

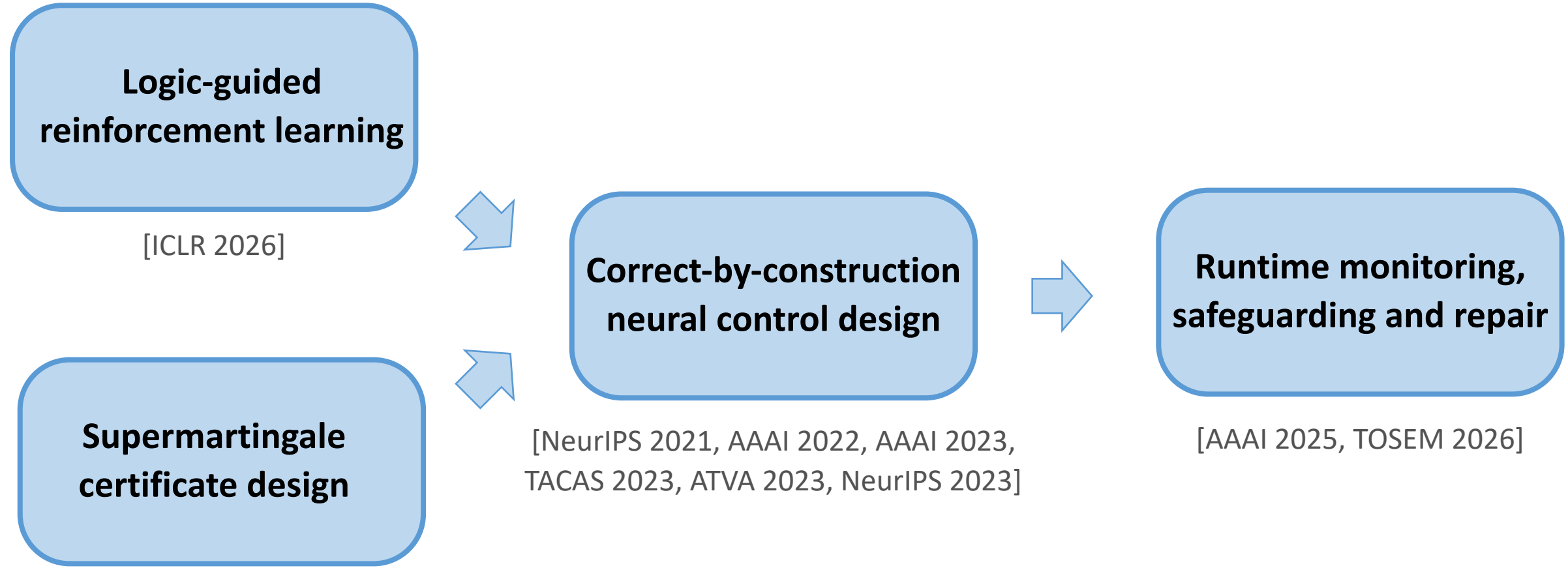
(SMT and bound propagation methods cannot handle probabilistic certificates)

Challenge 3

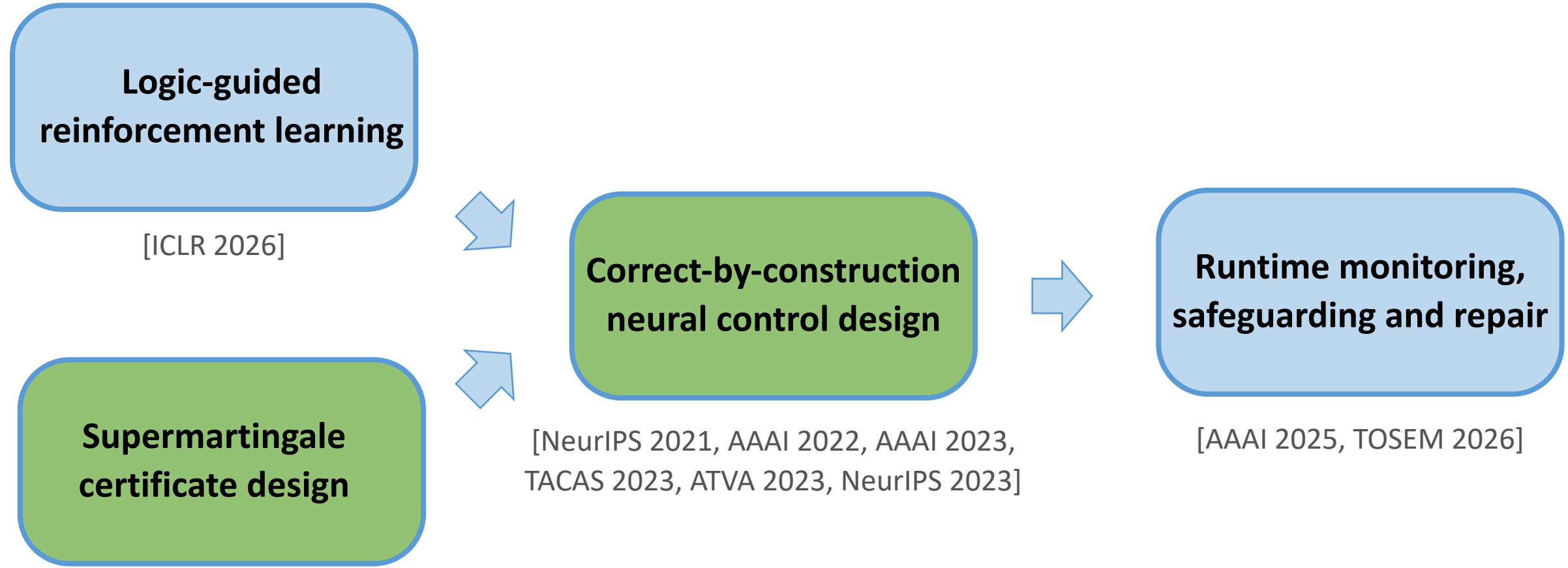
How to safeguard neural controllers upon deployment?

(Need to bridge the sim-to-reality gap)

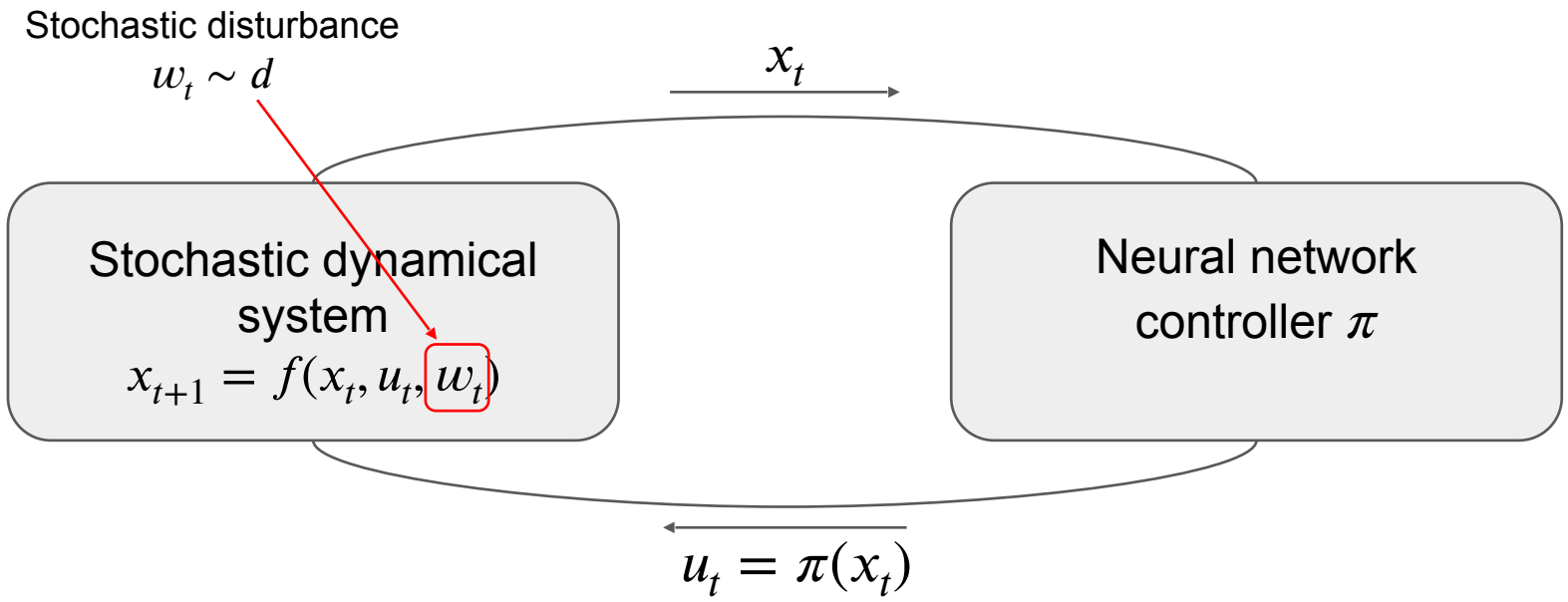
End-to-end pipeline for neural stochastic control with certificates



End-to-end pipeline for neural stochastic control with certificates



(discrete-time) Neural control verification problem



Given: Initial region X_0, temporal specification ϕ defining "good" traces, probability threshold $p \in [0,1]$

Control problem: Compute neural controller π that guarantee $\mathbb{P}^{\pi}_{x_0}[x_0, x_1, x_2, \ldots \vDash \phi] \geq p$ for all $x_0 \in X_0$

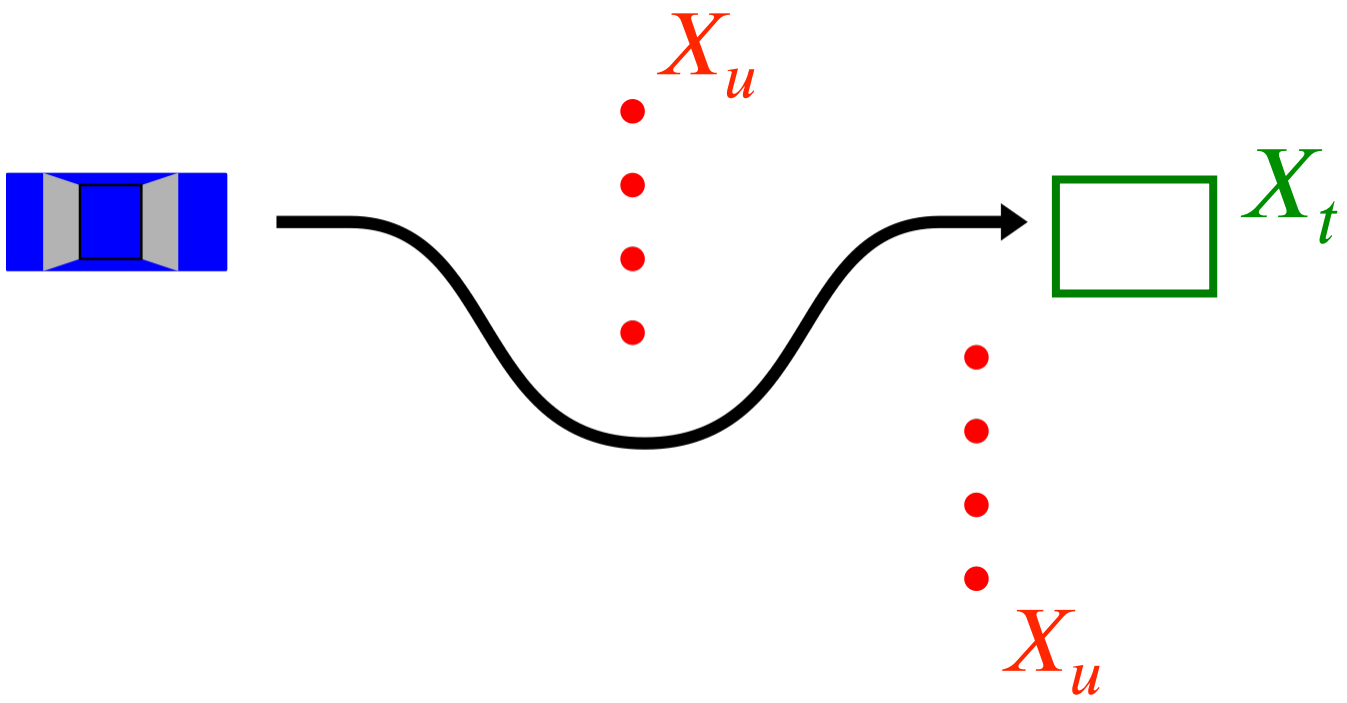
Verification problem: Verify that neural controller π guarantees $\mathbb{P}^{\pi}_{x_0}[x_0, x_1, x_2, \ldots \vDash \phi] \geq p$ for all $x_0 \in X_0$

What are supermartingales?

Supermartingale – stochastic process <u>decreasing</u> in expectation $\mathbb{E}[X_{n+1} \mid X_n] \leq X_n$

Martingale theory (Levý, 1934) based on many fundamental developments in mathematics

Example: Reach-avoid specifications



Reachability = reach the target set of states

Safety = do not reach the unsafe set of states

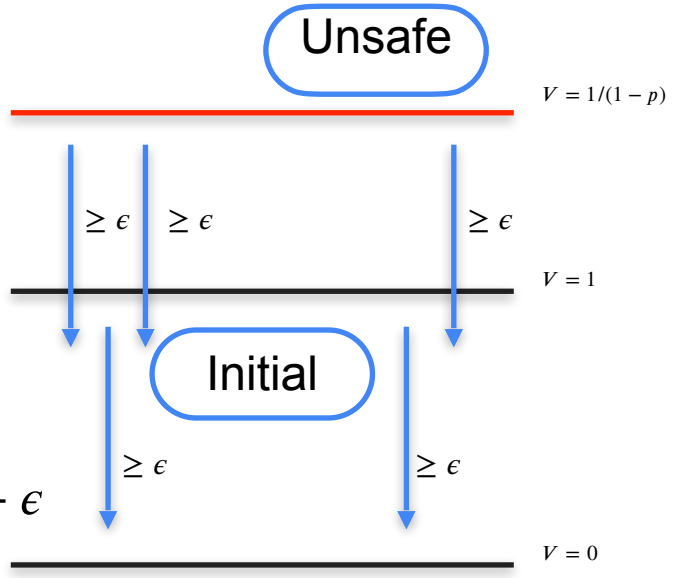
Reach-avoidance = reach the target set while avoiding the unsafe set of states

Reach-avoid supermartingale [AAAI'23]

(Joint work with M. Lechner, K. Chatterjee, T. A. Henzinger)

RASM is a measurable function $V: X \to \mathbb{R}$ such that:

1. Nonnegativity. $V(x) \geq 0$ for each $x \in X$.
2. Initial condition. $V(x) \leq 1$ for each initial state $x \in X_0$.
3. Safety condition. $V(x) \geq 1/(1-p)$ for each unsafe state $x \in X_u$.
4. Strict expected decrease. There exists $\epsilon > 0$ such that $V(x) \geq \mathbb{E}_{\omega \sim d}[V(f(x, \pi(x), \omega)] + \epsilon$ for $x \in X \backslash X_t$ at which $V(x) \leq 1/(1-p)$.

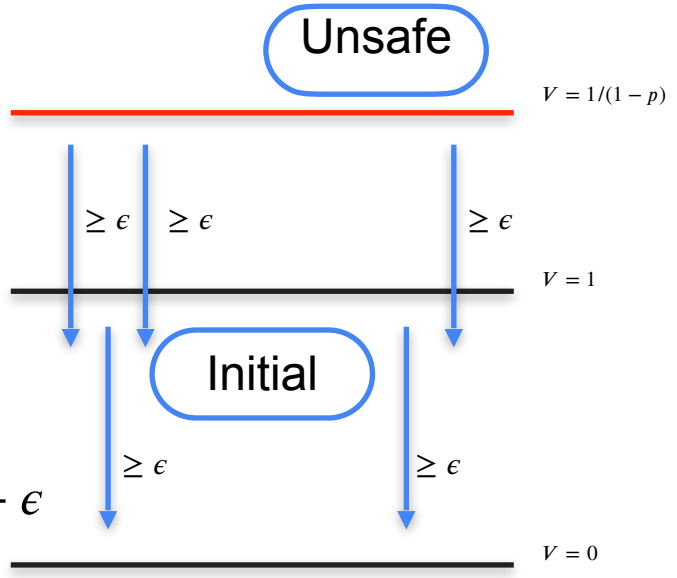


Reach-avoid supermartingale [AAAI'23]

(Joint work with M. Lechner, K. Chatterjee, T. A. Henzinger)

RASM is a measurable function $V: X \to \mathbb{R}$ such that:

1. Nonnegativity. $V(x) \geq 0$ for each $x \in X$.
2. Initial condition. $V(x) \leq 1$ for each initial state $x \in X_0$.
3. Safety condition. $V(x) \geq 1/(1-p)$ for each unsafe state $x \in X_u$.
4. Strict expected decrease. There exists $\epsilon > 0$ such that $V(x) \geq \mathbb{E}_{\omega \sim d}[V(f(x, \pi(x), \omega)] + \epsilon$ for $x \in X \backslash X_t$ at which $V(x) \leq 1/(1-p)$.



Theorem (Soundness). Suppose that the system admits a RASM. Then $\mathbb{P}^{\pi}_{x_0}[\text{ReachAvoid}(X_t, X_u)] \geq p$ for all $x \in X_0$.

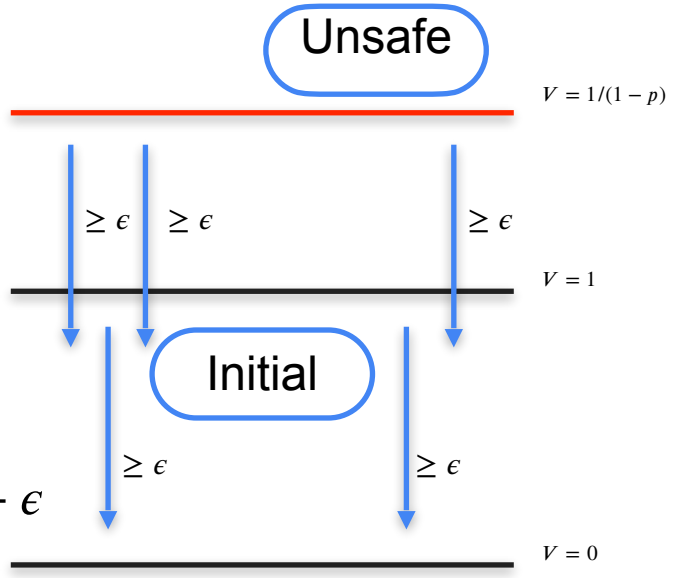
Reach-avoid supermartingale [AAAI'23]

(Joint work with M. Lechner, K. Chatterjee, T. A. Henzinger)

RASM is a measurable function $V: X \to \mathbb{R}$ such that:

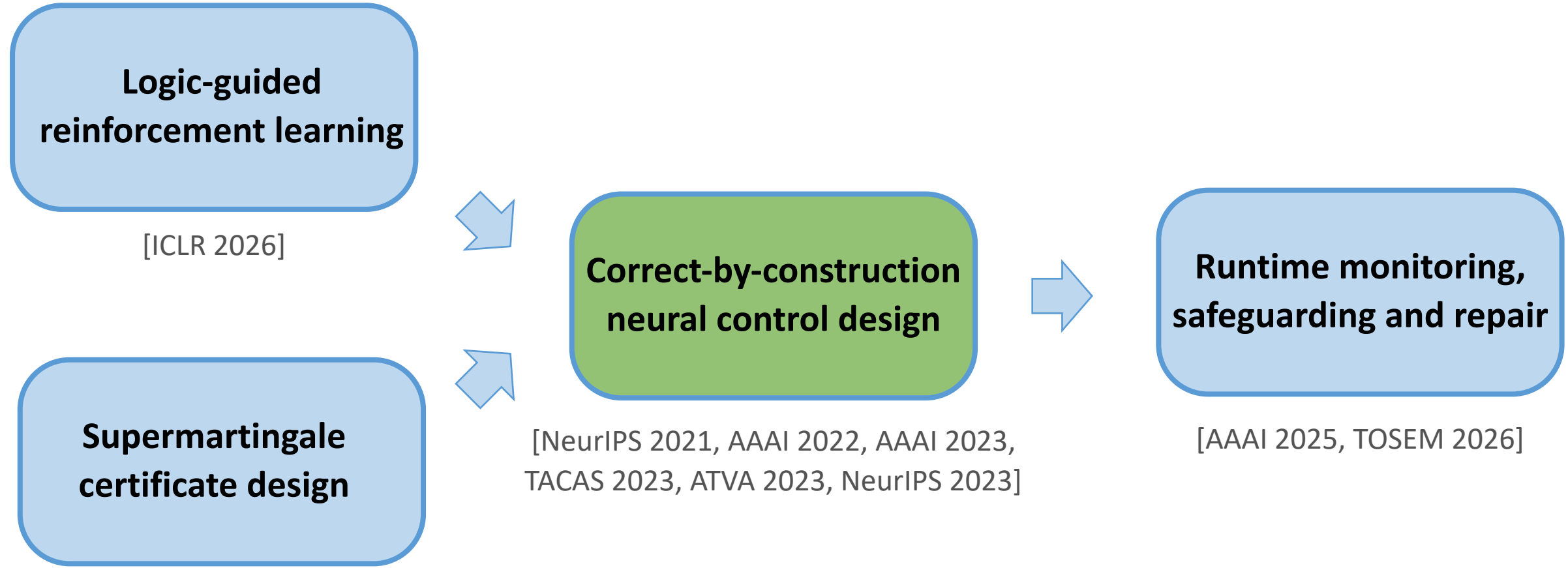
1. Nonnegativity. $V(x) \geq 0$ for
2. Initial condition
3. Safety condition $x \in X_u$.
4. Strict expected d $\epsilon > 0$ such that $V(x) \geq \mathbb{E}_{\omega \sim d}[V(f(x, \pi(x), \omega)] + \epsilon$ for $x \in X \backslash X_t$ at which $V(x) \leq 1/(1-p)$.

RASMs unify and generalize ranking supermartingales and stochastic barrier functions



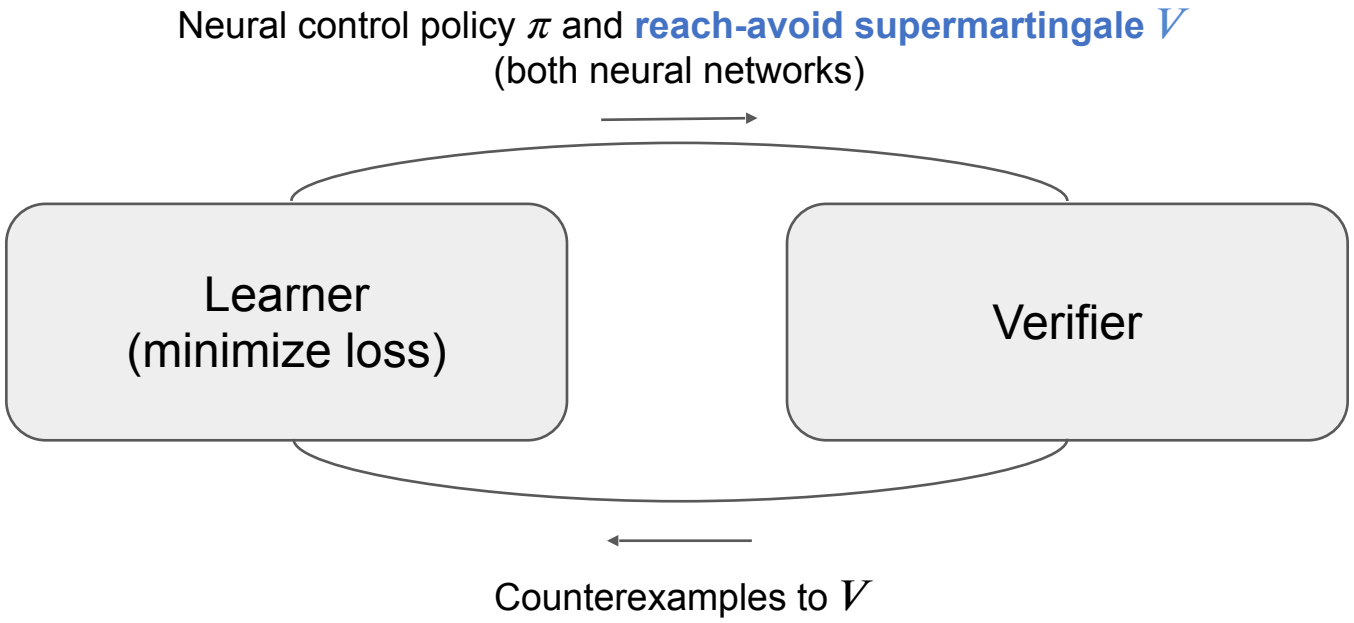
Theorem (Soundness). Suppose that the system admits a RASM. Then $\mathbb{P}^{\pi}_{x_0}[\text{ReachAvoid}(X_t, X_u)] \geq p$ for all $x \in X_0$.

End-to-end pipeline for neural stochastic control with certificates



Neural control with supermartingales [AAAI'22 & '23, TACAS'23]

(Joint work with M. Lechner, K. Chatterjee, T. A. Henzinger)

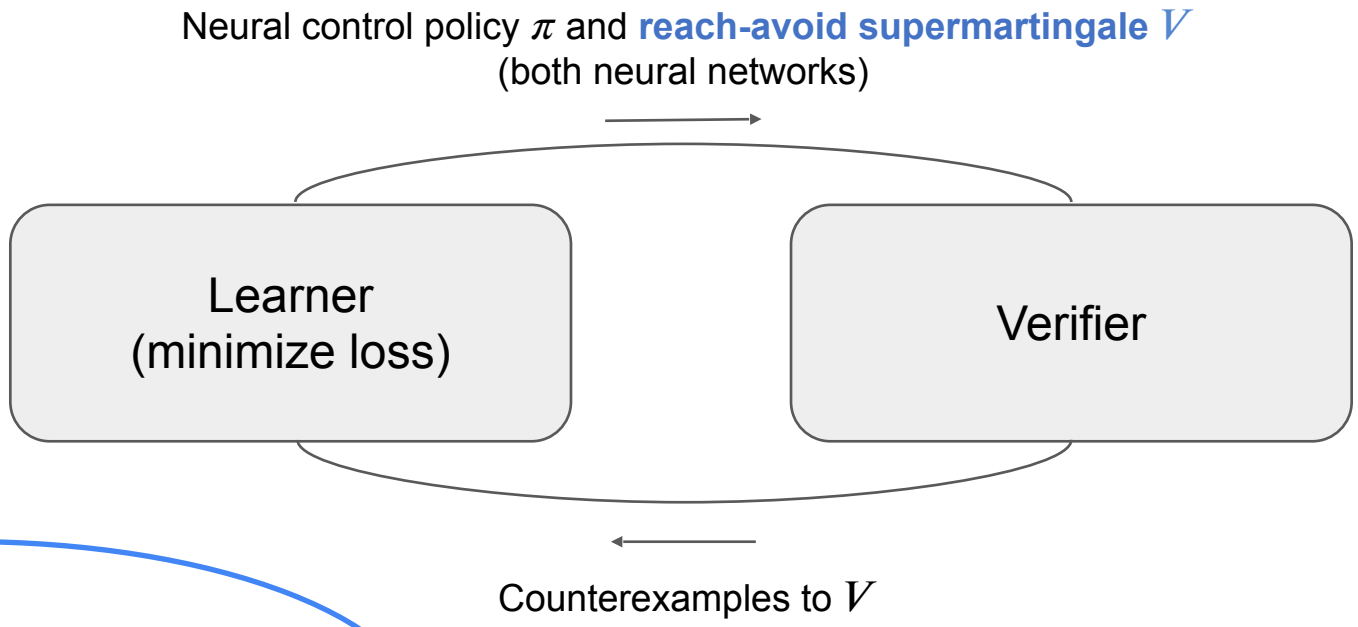


Assumptions (needed for automation):

(1) State space X of the system is compact

(2) Dynamics function f is (Lipschitz) continuous with Lipschitz constant L_f

Learner module



Loss function incurs loss if V_θ violates the RASM definition

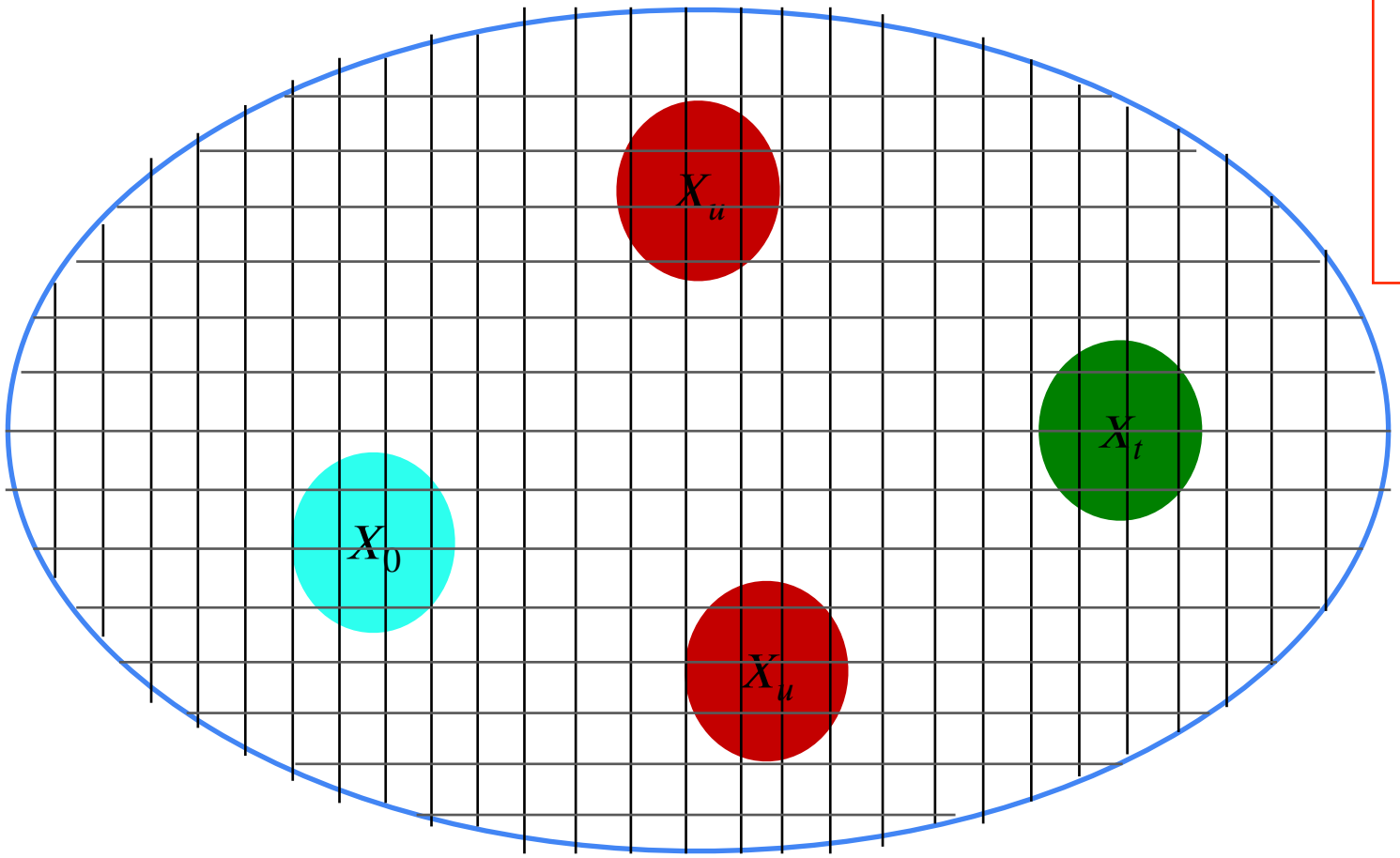
Learner module

Unsupervised learning task, loss function encodes defining conditions of RASMs

(Non-negativity imposed by default, by applying ReLU/softplus on neural RASM output)

$$\mathscr{L}(\theta, \nu) = \mathscr{L}_{\text{Init}}(\nu) + \mathscr{L}_{\text{Unsafe}}(\nu) + \mathscr{L}_{\text{Decrease}}(\theta, \nu)$$

Training set: Discretization



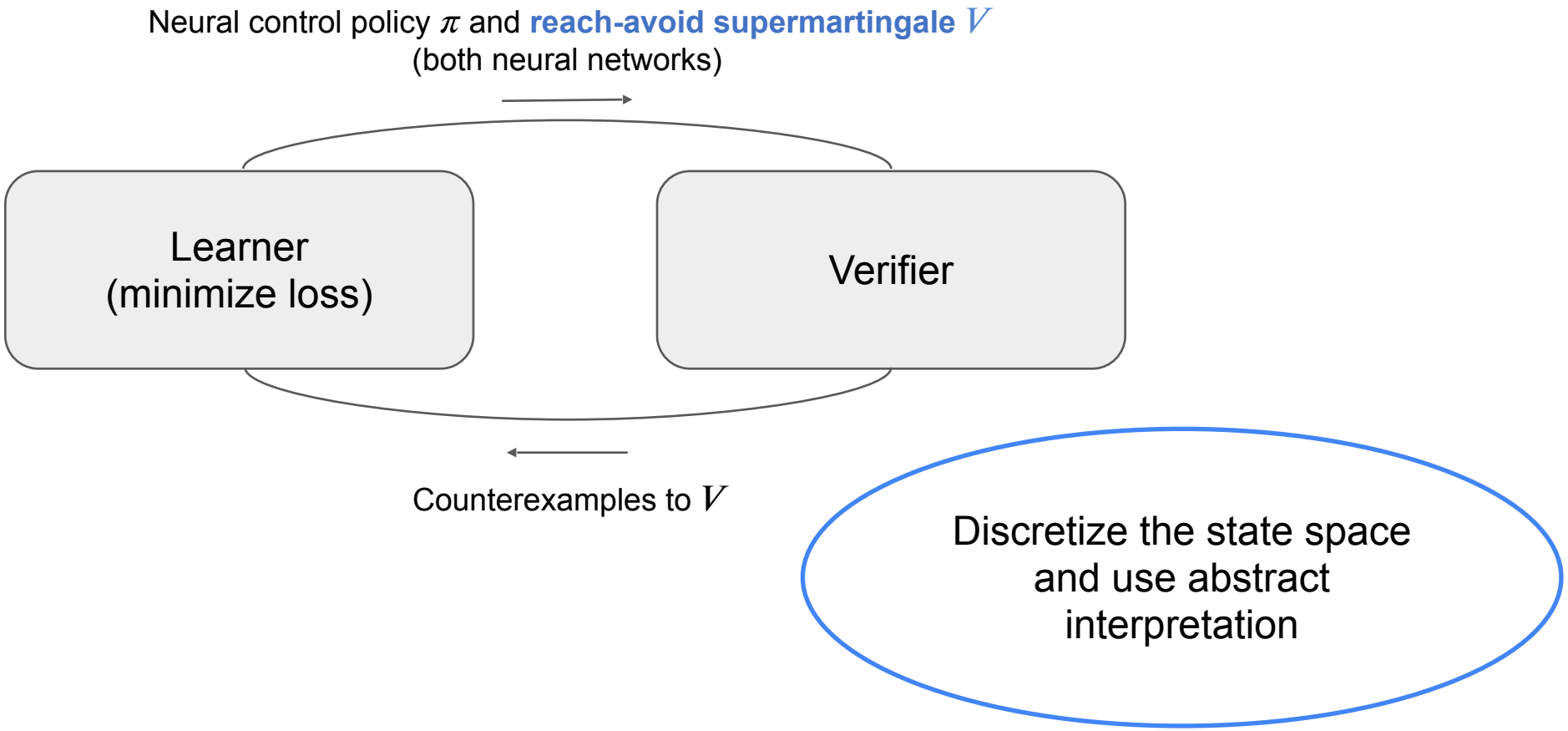
$\tilde{X}$ = hyperrectangular discretization of X

$C_{\text{init}} = X_0 \cap \tilde{X}$

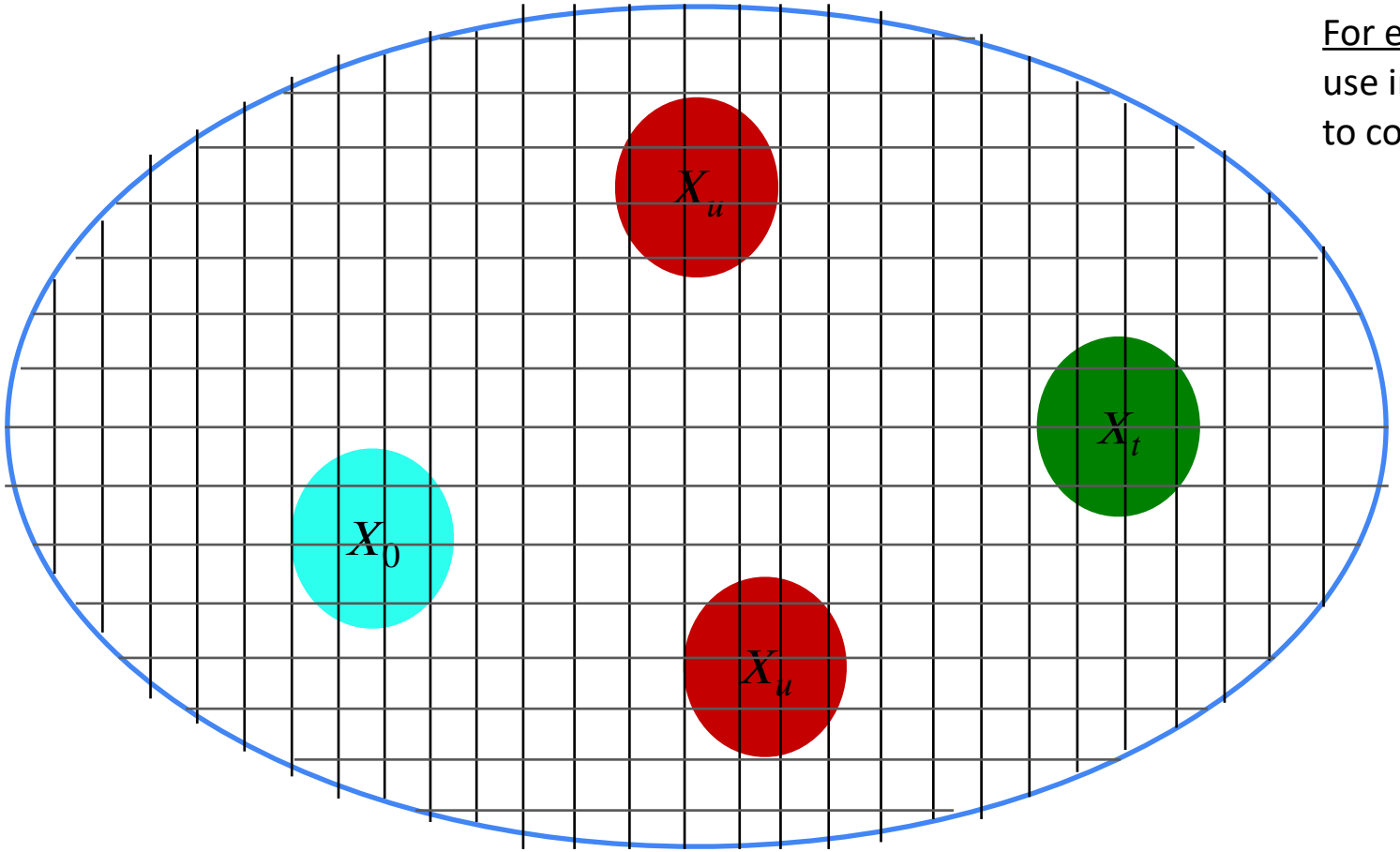
$C_{\text{unsafe}} = X_U \cap \tilde{X}$

$C_{\text{decrease}} = \tilde{X} \setminus (X_T \cup X_U)$

Verifier module



Verifier module

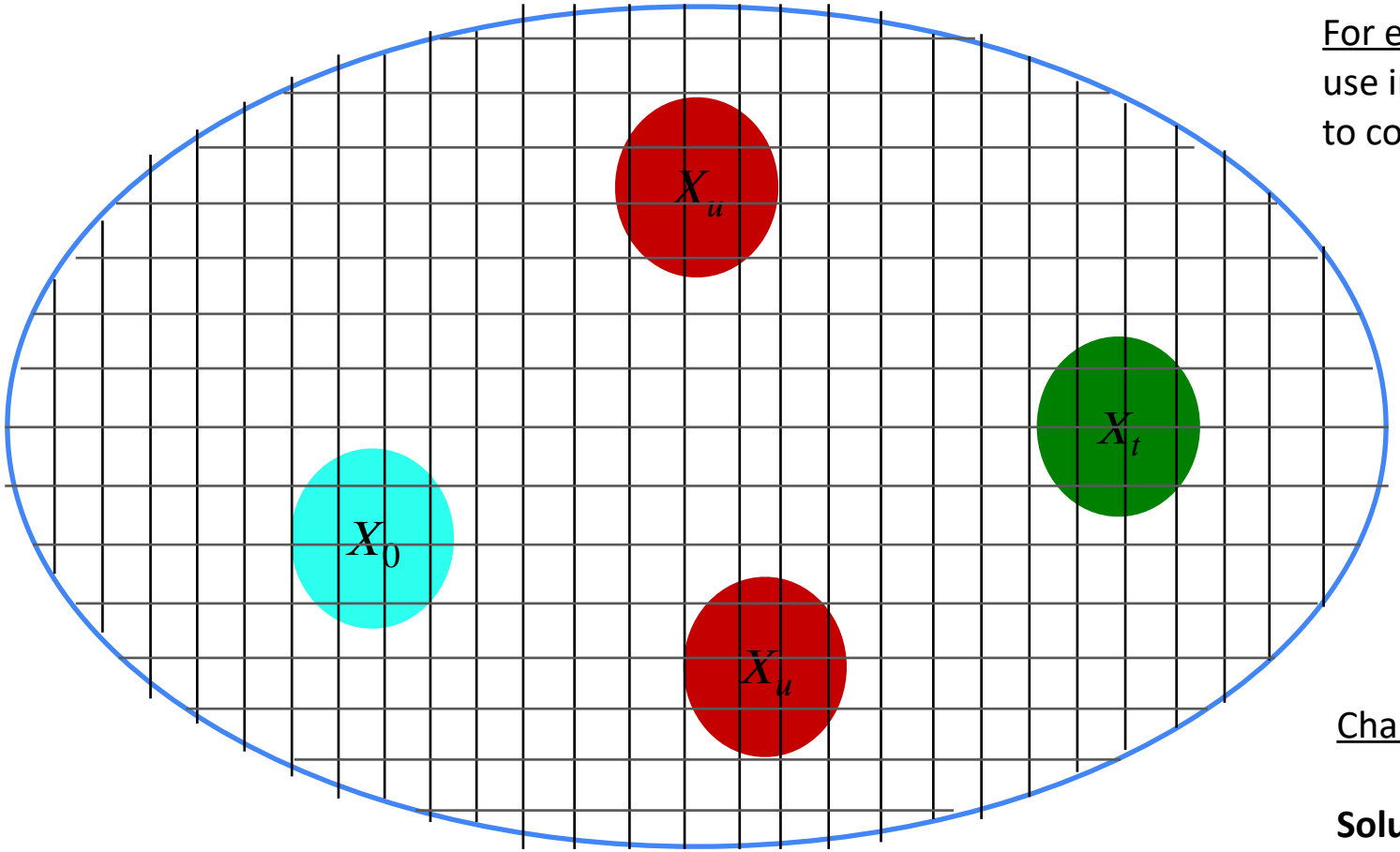


<u>For each discretization cell:</u>
use interval arithmetic abstract interpretation (IAAI) [1] to compute bounds on the RASM over each cell

Check Initial and Safety conditions of RASMs over all grid cells that intersect X_0 or X_u

[1] Gowal et al. *On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models.* arXiv 2018

Verifier module



For each discretization cell:
use interval arithmetic abstract interpretation (IAAI) [1] to compute bounds on the RASM over each cell

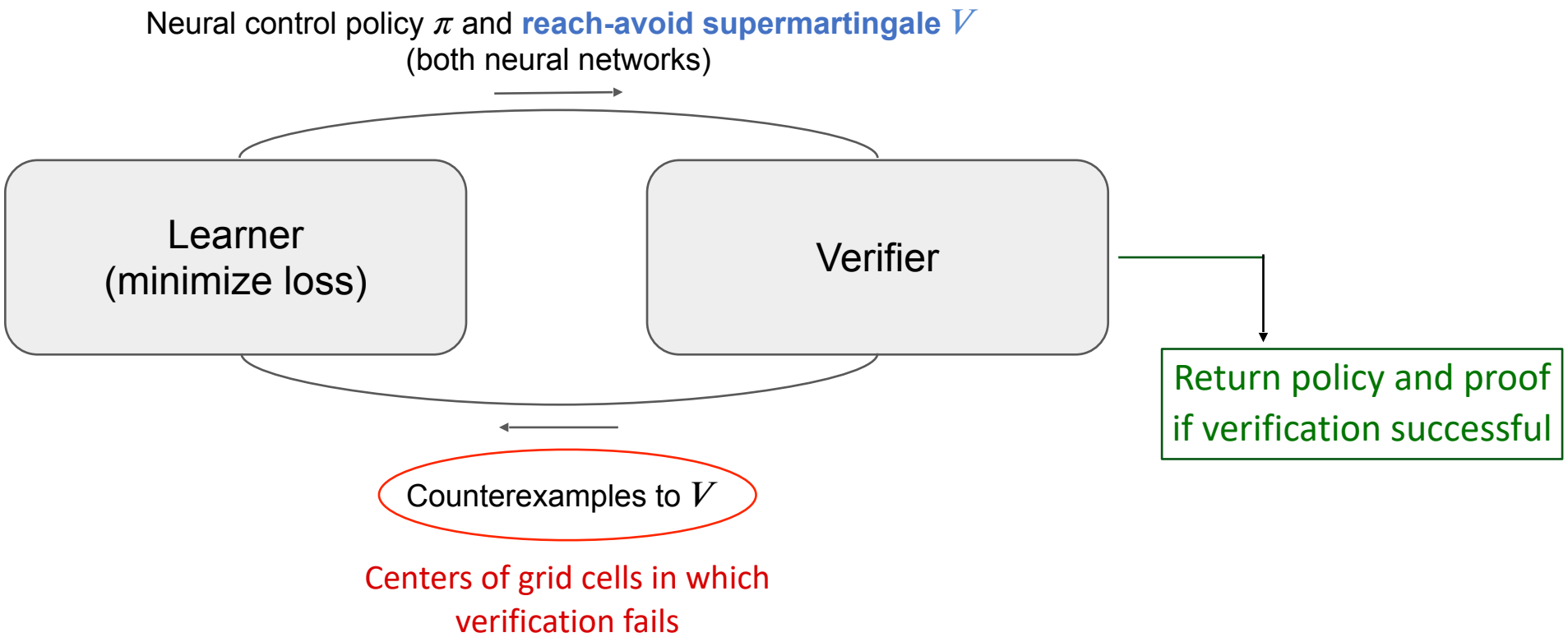
Check Initial and Safety conditions of RASMs over all grid cells that intersect X_0 or X_u

Challenge: How to verify expected decrease condition?

Solution: Lipschitz analysis + bound propagation!

[1] Gowal et al. *On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models.* arXiv 2018

Neural control with supermartingale certificates



Verifier guides the learner

1. **Counterexample guided inductive synthesis (CEGIS)**

   — counterexamples cell centers are added to training sets used by the learner

2. **Adaptive grid refinement**

   — grid cells that contain spurious counterexamples are refined

Experimental evaluation

- Initialize neural network policy using 100 iterations of PPO or SAC
- Policy network: [256,256] hidden dimension with ReLU activations
- RASM network: [256,256] hidden dimension with ReLU activations

Benchmark	Original tool (AAAI 2023)		With improvements	
	Probability	# iterations	Probability	# iterations
2D Linear	96.65	6	99.51	16
Inverted Pendulum	95.90	15	98.95	8
Collision avoidance	95.00	13	96.22	5
3D Linear	Fail	-	94.13	10
Humanoid.	Fail	-	69.44	32

See also: Badings et al. “Policy Verification in Stochastic Dynamical Systems Using Logarithmic Neural Certificates”. CAV 2025

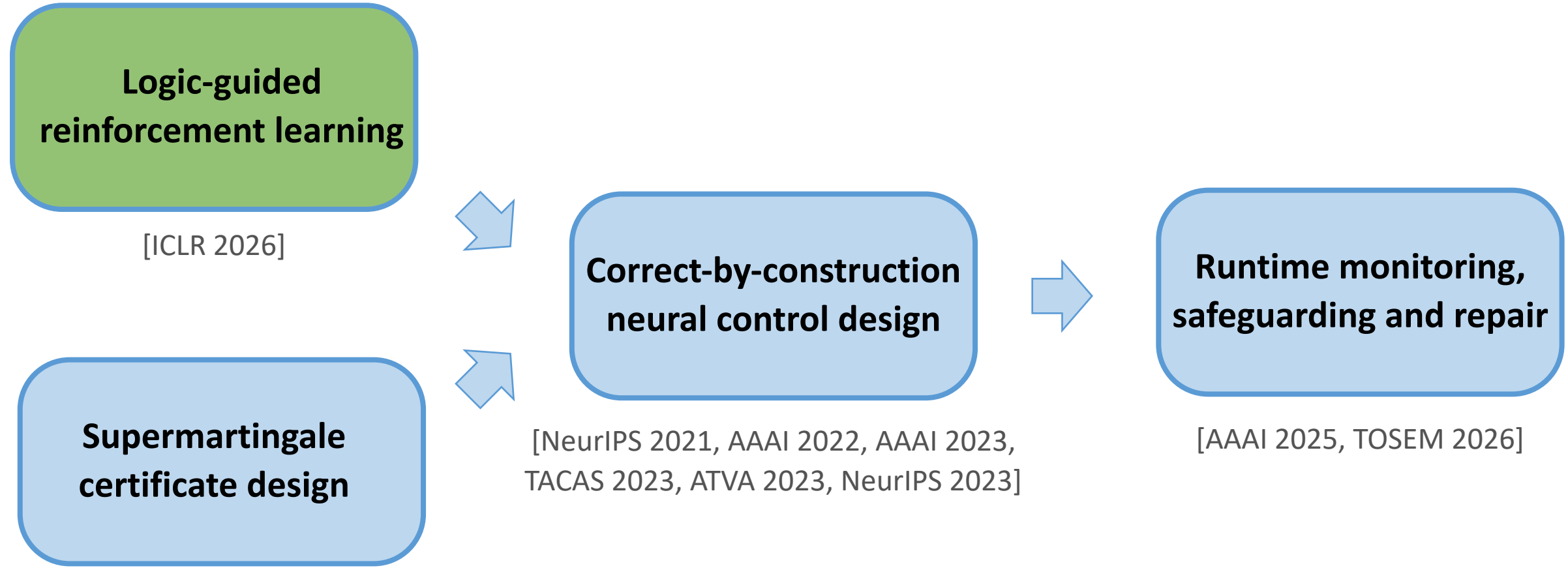
Extension to more general specifications

1. **Compositional reasoning** about reach-avoidance [NeurIPS'23]

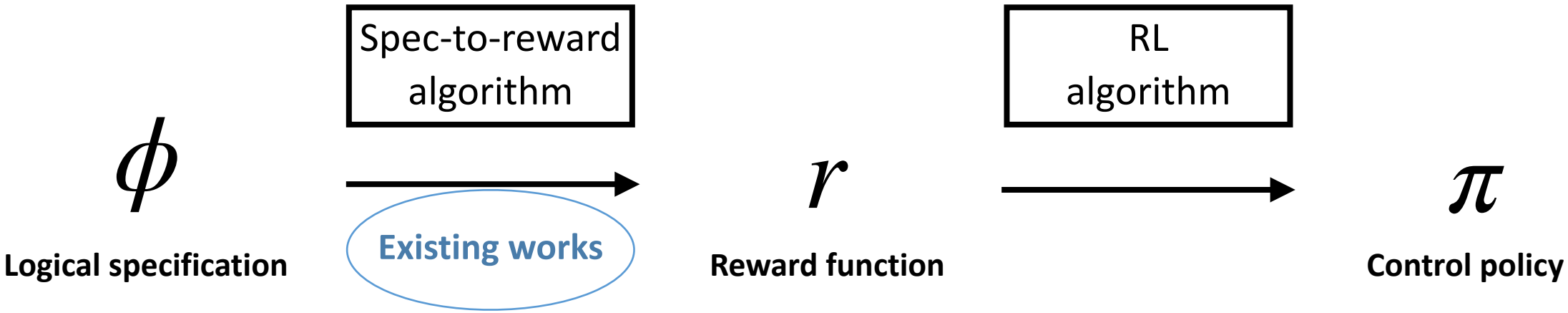
2. **Stability** (a.k.a. co-Büchi or reach-and-stay) with prob. 1 [ATVA'23]

3. Supermartingale certificates for **general omega-regular specifications** [CAV'25]

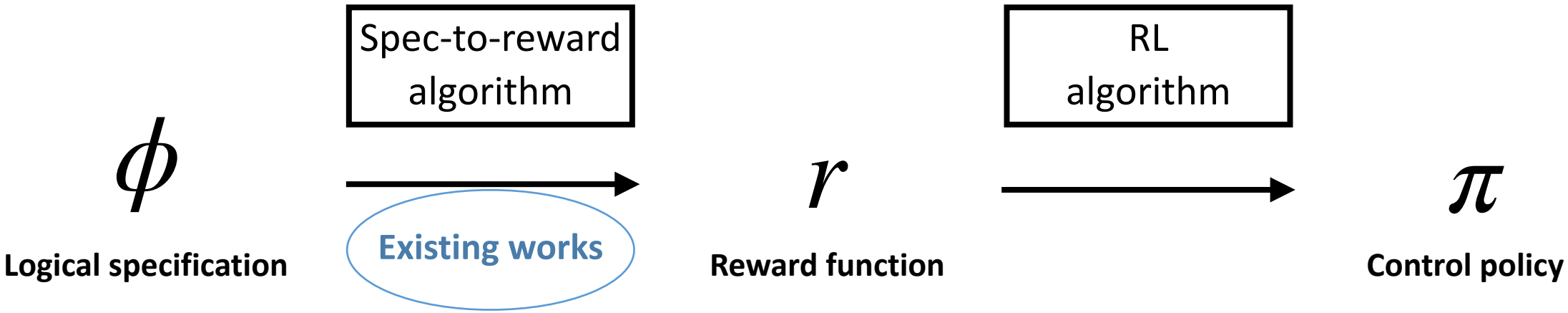
End-to-end pipeline for neural stochastic control with certificates



Logic-guided Reinforcement Learning



Logic-guided Reinforcement Learning



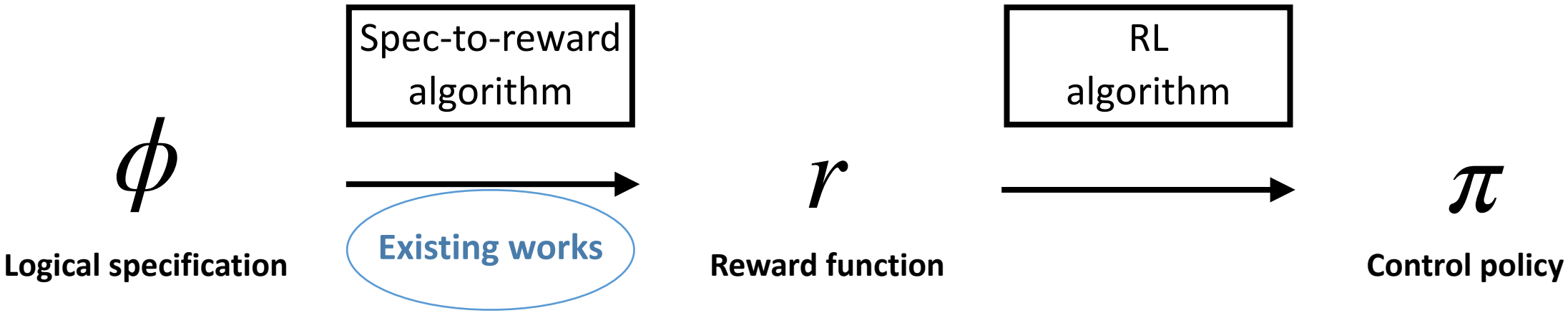
Challenge 1

Sparse reward function

Challenge 2

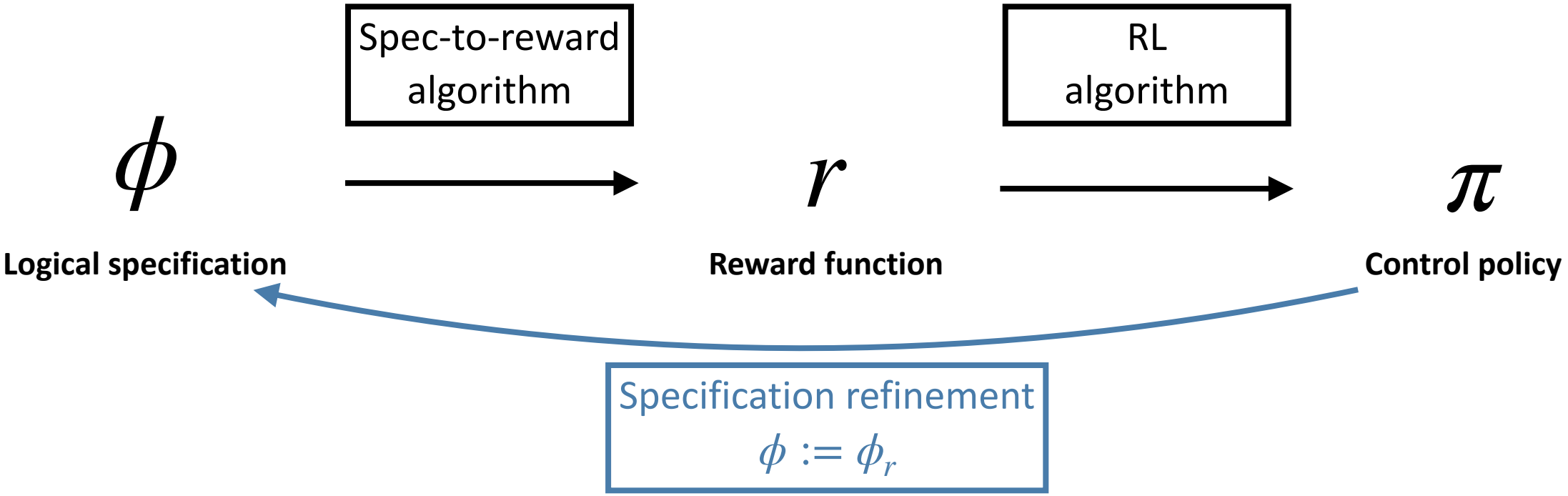
Coarse labelling function

Logic-guided Reinforcement Learning



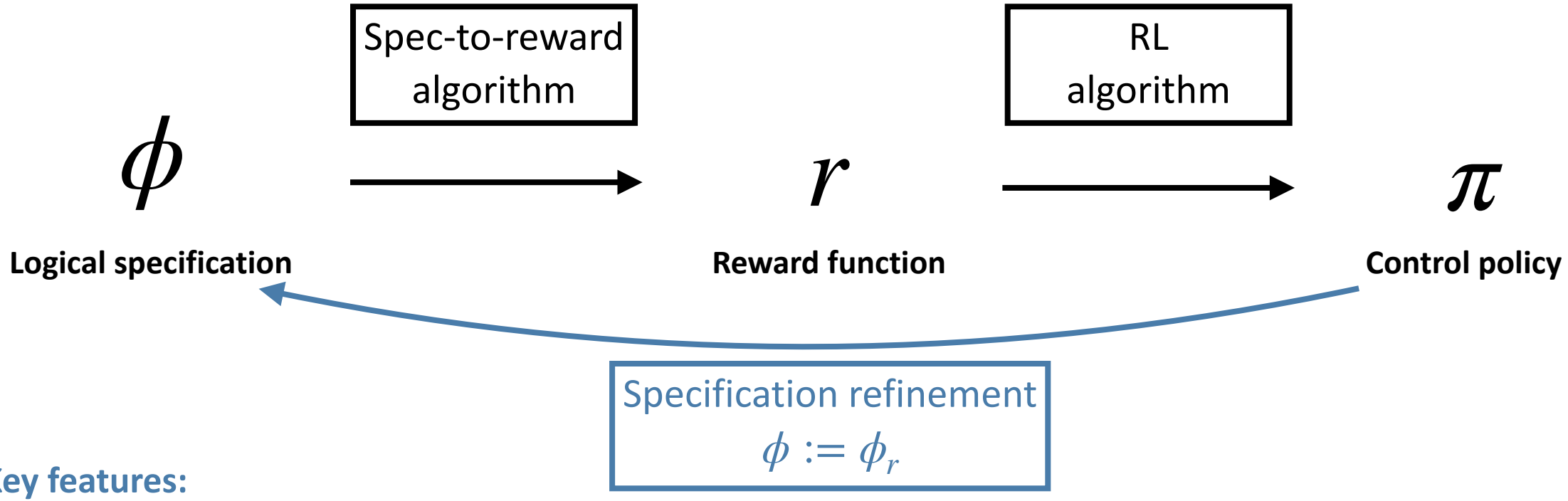
Why not refine the specification?

New Approach: Specification Refinement



AutoSpec: Automated Specification Refinement [ICLR'26]

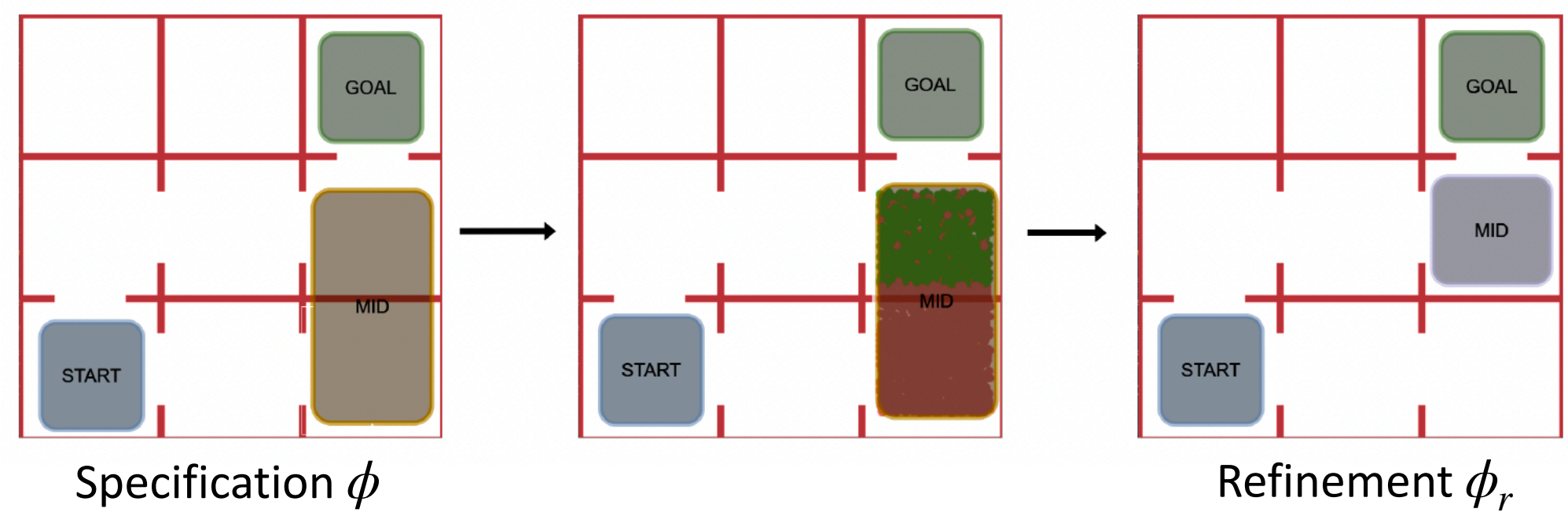
(Joint work with T. Ambadkar and A. Verma)



Key features:

1. **Fully automated, no manual specification tuning or reward engineering**
2. **Formal refinement guarantees**
3. **Applicable to SpectRL specifications**
4. **Compatible with any spec-to-reward algorithm for SpectRL**

Example of specification refinement

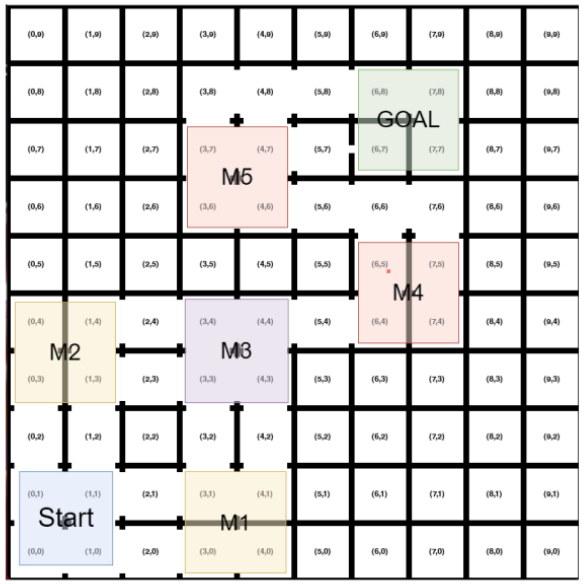


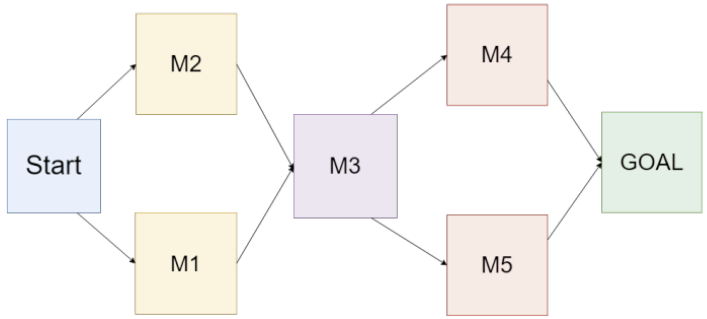
Specification ϕ

Refinement ϕ_r

Initial state in START, then reach a state in MID, then reach a state in GOAL

Experiment: 100-rooms Environment

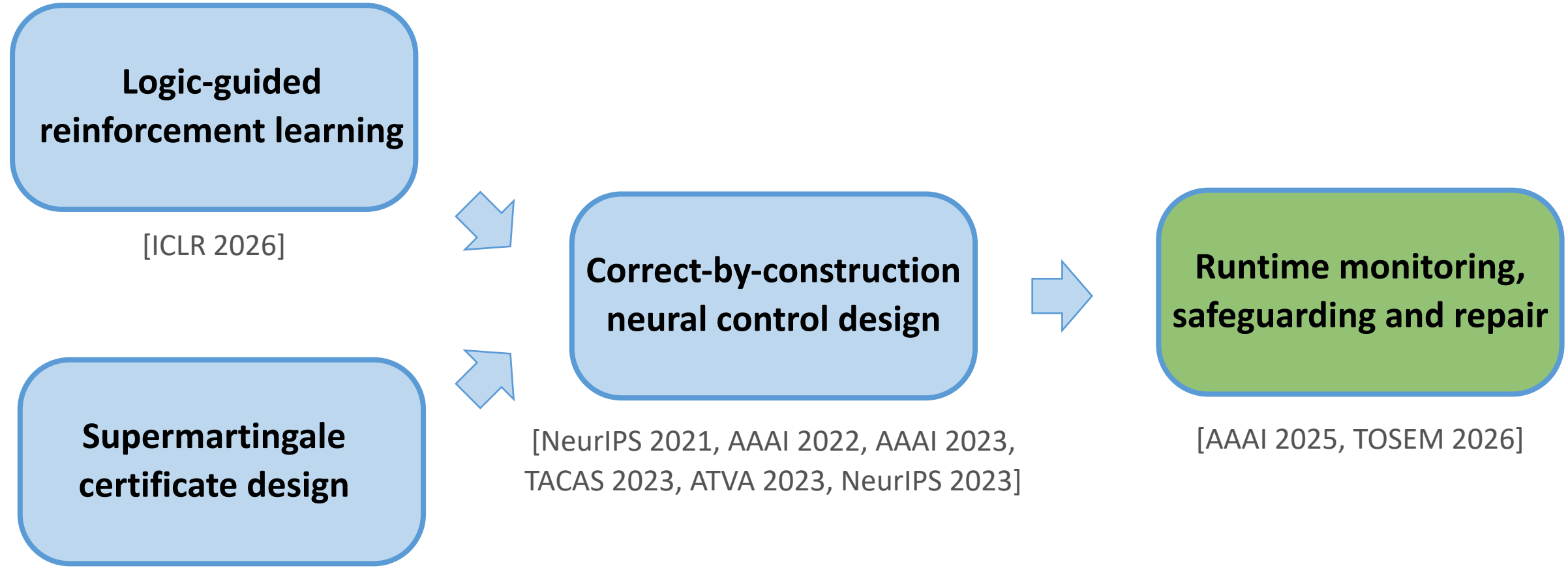




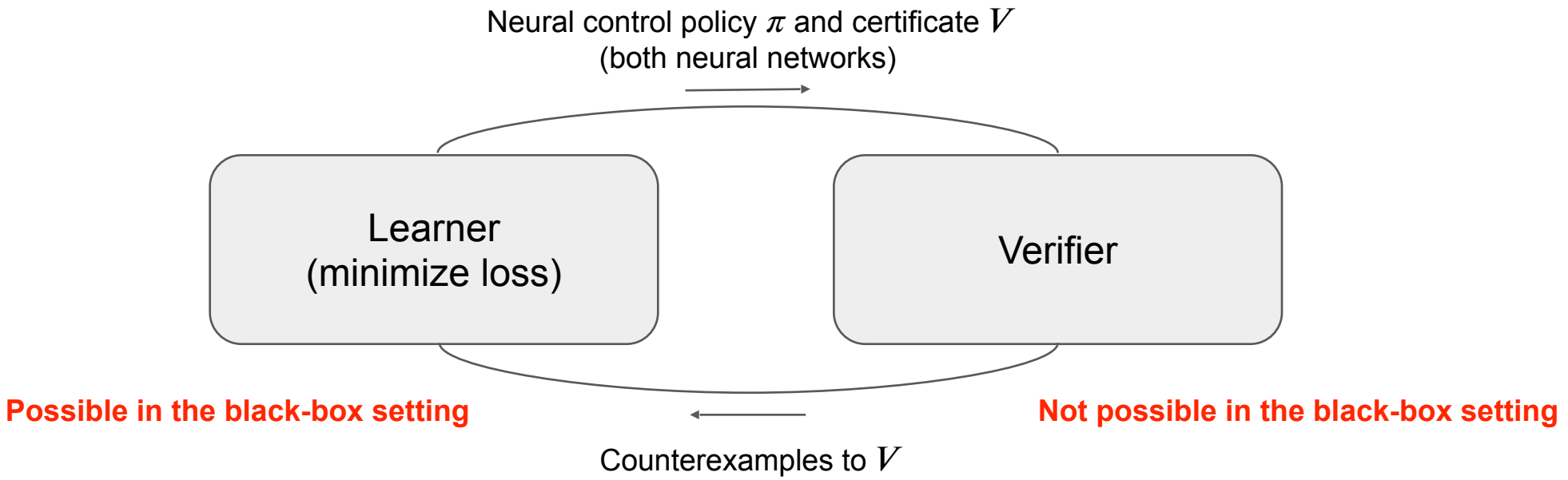
Comparison to DiRL [Jothimurugan et al., NeurIPS 2021]

- **Significant improvement** in satisfaction probability (~0% to >60%)
- ReachRefine applied to START - M1
- PastRefine applied to M1 - M3 and M2 - M3
- OrRefine applied to M4 - GOAL

End-to-end pipeline for neural stochastic control with certificates



How to bridge the sim-to-reality gap?

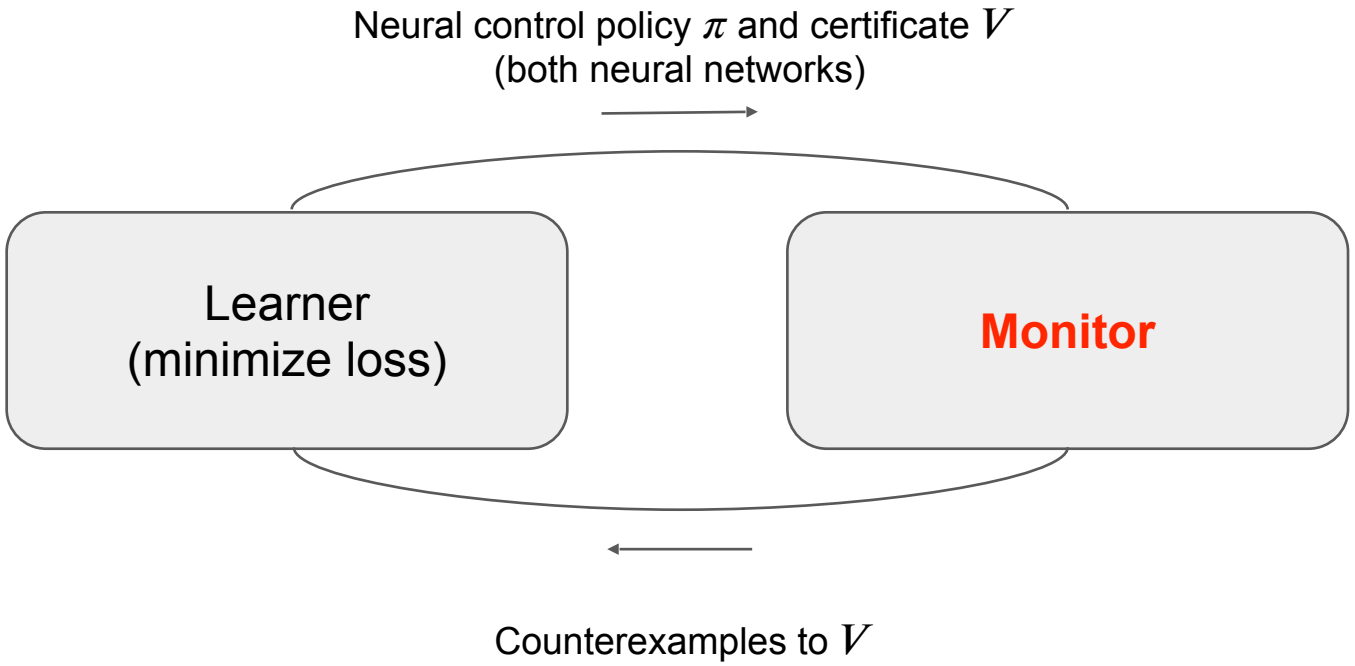


How to ensure correctness of learned controllers and certificates upon deployment?

How to repair them if they are incorrect?

Runtime monitoring in black-box systems [AAAI'25]

(Joint work with E. Yu and T. A. Henzinger)



Monitor certificate condition violations

➡ Use **certificate violations** as new training data

➡ **Repair** by re-learning

Runtime monitoring-guided repair

	$\#D_{\text{NEW}}$	**SR** (%)	**BR** (%)	**NDR** (%)
Initialized	-	93.99	87.03	45.38
Baseline	878	96.61	-	-
CertPM	548	**99.13**	**100.00**	90.66
PredPM [0,0,-1]	146	99.06	**100.00**	90.11
PredPM [0,1,-5]	355	98.67	**100.00**	90.12
PredPM [2,2,0]	**1000**	99.09	**100.00**	**91.67**

Drone Environment [1]

(Navigate a drone among 1024 other drones)

8D encoding

Initialized: Neural controller and barrier certificate learned via SABLAS [1]

Baseline: Monitor and repair only with hard safety violations

CertPM + PredPM: Monitor and repair with hard safety violations + certificate violations

[1] Qin et al. “SABLAS: Learning Safe Control for Black-box Dynamical Systems”. IEEE Robotics and Automation Letters 2022

Future directions

- Neural control under richer specifications (omega-regular specifications) [CAV'25]
- Scalability challenge
- Runtime monitoring and shielding of neural controllers [AAAI'25, TOSEM'26]
- Continuous-time control (see Neustroev et al., AAAI 2025)

Open positions

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

❗ Multiple **PhD positions** (fully funded)

❗ Multiple **Visiting Research Student** positions, 6 months

Possible topic include (but not limited to):

- Formal verification and synthesis in Markov models
- Probabilistic program verification
- Certification of neural control systems
- Runtime monitoring and safeguarding
- Safe reinforcement learning
- Runtime monitoring and safeguarding of LLM agents

Contact: djordje.zikelic@ntu.edu.sg

More details: https://djordjezikelic.github.io/openings/

End-to-end pipeline for neural stochastic control with certificates